```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
```

```
data = pd.read_excel("Iris_train.xlsx")
```

```
data.head()
```

	Sepal-length	sepal-width	petal-length	petal-width	class
0	5.0	2.0	3.5	1.0	Iris-versicolor
1	6.0	2.2	4.0	1.0	Iris-versicolor
2	6.0	2.2	5.0	1.5	Iris-virginica
3	6.2	2.2	4.5	1.5	Iris-versicolor
4	4.5	2.3	1.3	0.3	Iris-setosa

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
x = data.iloc[:,:-1]  #selecting all rows and all columns excpet the last as it is
x
```

	Sepal-length	sepal-width	petal-length	petal-width
0	5.0	2.0	3.5	1.0
1	6.0	2.2	4.0	1.0
2	6.0	2.2	5.0	1.5
3	6.2	2.2	4.5	1.5
4	4.5	2.3	1.3	0.3
...	...	...	...	...
115	5.4	3.9	1.3	0.4
116	5.8	4.0	1.2	0.2
117	5.2	4.1	1.5	0.1
118	5.5	4.2	1.4	0.2
119	5.7	4.4	1.5	0.4

120 rows × 4 columns

Next steps: Generate code with x | View recommended plots | New interactive sheet

```
#printing y i.e labels
y = data.iloc[:,-1]   #all rows but only last column
y
```

	class
0	Iris-versicolor
1	Iris-versicolor
2	Iris-virginica
3	Iris-versicolor
4	Iris-setosa
...	...
115	Iris-setosa
116	Iris-setosa
117	Iris-setosa
118	Iris-setosa
119	Iris-setosa

120 rows × 1 columns

dtype: object

```
#training the model
model = RandomForestClassifier()
model.fit(x.values,y.values)
```

```
▾ RandomForestClassifier
RandomForestClassifier()
```

```
predictions = model.predict([[6.2,2.8,4.8,1.8]])
predictions
```

```
array(['Iris-virginica'], dtype=object)
```

```
test_data = pd.read_excel("Iris_test.xlsx")
```

```
x_test = test_data.iloc[:,:-1]
```

```
test_result = model.predict(x_test.values)
test_result
```

```
array(['Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor'], dtype=object)
```

```
y_test = test_data.iloc[:,-1]
y_test.values
```

```
array(['Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-virginica', 'Iris-versicolor', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-virginica', 'Iris-virginica', 'Iris-virginica',
       'Iris-setosa', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-versicolor',
       'Iris-versicolor', 'Iris-versicolor', 'Iris-virginica',
       'Iris-virginica', 'Iris-setosa', 'Iris-setosa', 'Iris-setosa',
       'Iris-setosa', 'Iris-setosa', 'Iris-setosa', 'Iris-versicolor',
       'Iris-versicolor'], dtype=object)
```

Accuracy tells us how often the model was right

```
actual = y_test.values
```

```
count = 0
for i in range(0,len(actual)):
    if actual[i] == test_result[i]:
        count +=1
accuracy = (count*100)/len(actual)
print(f"The accuracy of the above model is {accuracy}!")
```

```
The accuracy of the above model is 96.66666666666667!
```